

FLUTTER

Durum ve Formlar

9–12. haftalar

BT2025MOB1106 modülünün kazanımlarından hareketle; Android/Java ayrıntılarını kopyalamadan, öğrenci dostu öğrenme sırasıyla hazırlanmıştır.

9 birim • 36 hafta • 72 çalışan Flutter uygulaması • üç seviyeli uygulama görevleri

Öğrenme Haritası

Hafta	Ders	Uygulamalar
9	StatelessWidget ve StatefulWidget	Sayaç • Pomodoro Sayacı
10	Butonlar ve Olaylar	Renk Değiştirici • Alışkanlık İşaretleyici
11	Form ve Doğrulama	Kayıt Formu • Etkinlik Başvurusu
12	Seçim Bileşenleri	Tercih Formu • Kantin Siparişi

3. Birim — Durum ve Formlar

Bu birimde dört haftalık öğrenme döngüsü kullanılır: gör, birlikte çöz, değiştir, yeni probleme aktar.

9. Hafta — StatelessWidget ve StatefulWidget

Sabit ve değişen arayüzleri ayırma, setState

Bu derste konu, önce çalışan bir mobil arayüz üzerinde görülür; sonra widget ağacı küçük parçalara ayrılır. Kaynak kitaptaki kazanım korunur, ancak XML/Activity merkezli sıra yerine Flutter'ın widget, durum ve geri bildirim yaklaşımı kullanılır.

1	Sabit ve değişen arayüzleri ayırma, setState
2	Girdi, işlem, çıktı ve durum değişimini ayırt etme
3	Okunabilir Dart kodu ve yeniden kullanılabilir widget geliştirme
4	Boş, sınır ve hatalı verilerle test etme

Çözümlü kitap uyarlaması: Sayaç

1. MaterialApp ve Scaffold ile ekran iskeletini kurun. 2. Veriyi TextField ile alın. 3. Değişen değeri State içinde tutun. 4. Olaydan sonra setState ile ekranı yenileyin. 5. Kullanıcıya açık sonuç ve hata mesajı gösterin.

```
FilledButton(onPressed: () => setState(() => value++), child: const Text('Uygula'));
```

Çalışan ekran — Sayaç

Sayaç

HAFTA 9

Sayaç

Sabit ve değişen arayüzleri ayırma, setState



Deneme verisi

► Uygula

Sıfırla

9

Canlı sonuç

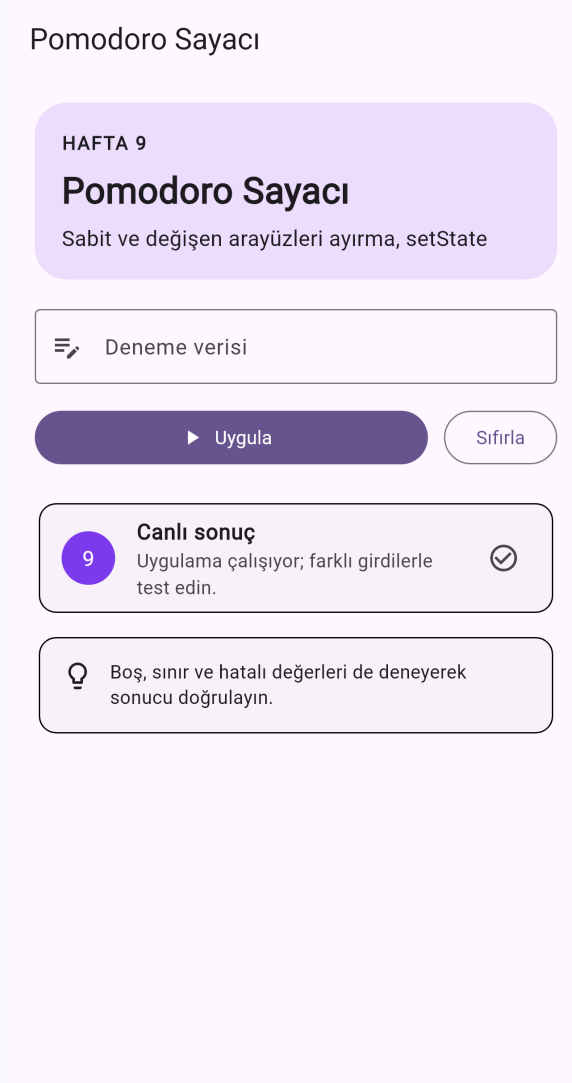
Uygulama çalışıyor; farklı girdilerle test edin.



Boş, sınır ve hatalı değerleri de deneyerek sonucu doğrulayın.

Bu ekran Flutter'ın gerçek widget testinden alınmıştır; kaynak proje çevrimdışı pakette çalışır durumdadır.

Özgün günlük yaşam uygulaması — Pomodoro Sayacı



Aynı teknik kazanım yeni bir problem bağlamında uygulanır. Öğrenci, örneği kopyalamak yerine veriyi ve ekran geri bildirimini kendi senaryosuna göre değiştirir.

Uygulama zamanı

Kolay	Başlık, renk, simge ve açıklamayı kendi senaryonuza uyarlayın.
Orta	Boş ve geçersiz girdiyi engelleyip Türkçe doğrulama mesajı gösterin.
İleri	Sonuçları bir modele dönüştürüp ListView içinde geçmiş olarak saklayın; silme ekleyin.

Geri bildirimden sonra kontrol: controller kaynakları dispose edildi mi, dar ekranda taşma var mı, boş/hata/yükleniyor durumları gösteriliyor mu ve en az üç test verisi denendi mi?

10. Hafta — Butonlar ve Olaylar

ElevatedButton, IconButton ve kullanıcı olayları

Bu derste konu, önce çalışan bir mobil arayüz üzerinde görülür; sonra widget ağacı küçük parçalara ayrılır. Kaynak kitaptaki kazanım korunur, ancak XML/Activity merkezli sıra yerine Flutter'ın widget, durum ve geri bildirim yaklaşımı kullanılır.

1	ElevatedButton, IconButton ve kullanıcı olayları
2	Girdi, işlem, çıktı ve durum değişimini ayırt etme
3	Okunabilir Dart kodu ve yeniden kullanılabilir widget geliştirme
4	Boş, sınır ve hatalı verilerle test etme

Çözümlü kitap uyarlaması: Renk Değiştirici

1. MaterialApp ve Scaffold ile ekran iskeletini kurun. 2. Veriyi TextField ile alın. 3. Değişen değeri State içinde tutun. 4. Olaydan sonra setState ile ekranı yenileyin. 5. Kullanıcıya açık sonuç ve hata mesajı gösterin.

```
FilledButton(onPressed: () => setState(() => value++), child: const Text('Uygula'));
```

Çalışan ekran — Renk Değiştirici

Renk Değiştirici

HAFTA 10

Renk Değiştirici

ElevatedButton, IconButton ve kullanıcı olayları



Deneme verisi

► Uygula

Sıfırla

10

Canlı sonuç

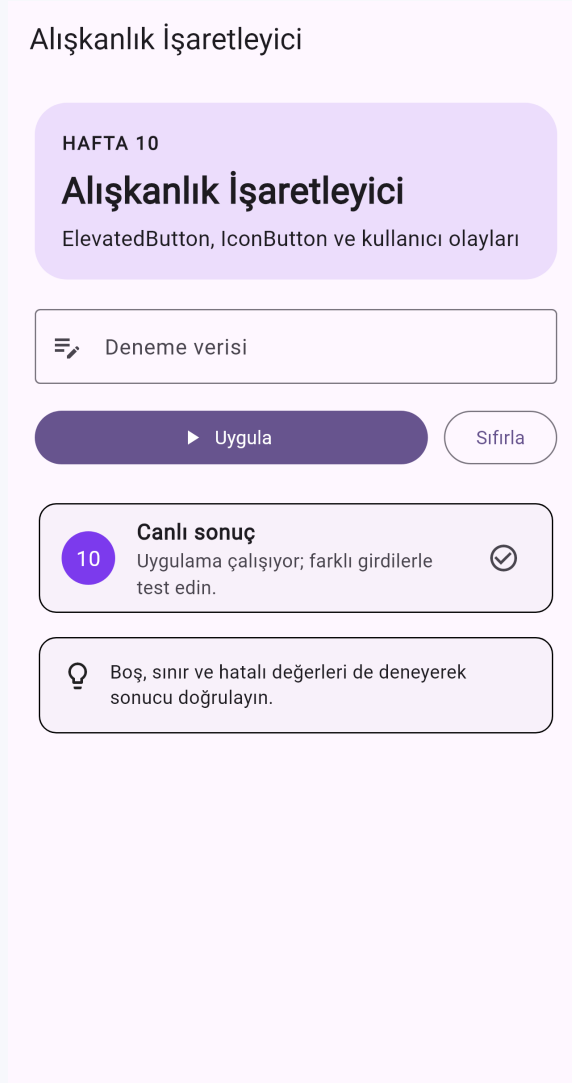
Uygulama çalışıyor; farklı girdilerle test edin.



Boş, sınır ve hatalı değerleri de deneyerek sonucu doğrulayın.

Bu ekran Flutter'ın gerçek widget testinden alınmıştır; kaynak proje çevrimdışı pakette çalışır durumdadır.

Özgün günlük yaşam uygulaması — Alışkanlık İşaretleyici



Aynı teknik kazanım yeni bir problem bağlamında uygulanır. Öğrenci, örneği kopyalamak yerine veriyi ve ekran geri bildirimini kendi senaryosuna göre değiştirir.

Uygulama zamanı

Kolay	Başlık, renk, simge ve açıklamayı kendi senaryonuza uyarlayın.
Orta	Boş ve geçersiz girdiyi engelleyip Türkçe doğrulama mesajı gösterin.
İleri	Sonuçları bir modele dönüştürüp ListView içinde geçmiş olarak saklayın; silme ekleyin.

Geri bildirimden sonra kontrol: controller kaynakları dispose edildi mi, dar ekranda taşma var mı, boş/hata/yükleniyor durumları gösteriliyor mu ve en az üç test verisi denendi mi?

11. Hafta — Form ve Doğrulama

Form, TextFormField, validator ve güvenli veri alma

Bu derste konu, önce çalışan bir mobil arayüz üzerinde görülür; sonra widget ağacı küçük parçalara ayrılır. Kaynak kitaptaki kazanım korunur, ancak XML/Activity merkezli sıra yerine Flutter'ın widget, durum ve geri bildirim yaklaşımı kullanılır.

1	Form, TextFormField, validator ve güvenli veri alma
2	Girdi, işlem, çıktı ve durum değişimini ayırt etme
3	Okunabilir Dart kodu ve yeniden kullanılabilir widget geliştirme
4	Boş, sınır ve hatalı verilerle test etme

Çözümlü kitap uyarlaması: Kayıt Formu

1. MaterialApp ve Scaffold ile ekran iskeletini kurun. 2. Veriyi TextField ile alın. 3. Değişen değeri State içinde tutun. 4. Olaydan sonra setState ile ekranı yenileyin. 5. Kullanıcıya açık sonuç ve hata mesajı gösterin.

```
FilledButton(onPressed: () => setState(() => value++), child: const Text('Uygula'));
```

Çalışan ekran — Kayıt Formu

Kayıt Formu

HAFTA 11

Kayıt Formu

Form, TextFormField, validator ve güvenli veri alma

☰ Deneme verisi

▶ Uygula

Sıfırla

11

Canlı sonuç

Uygulama çalışıyor; farklı girdilerle test edin.



Boş, sınır ve hatalı değerleri de deneyerek sonucu doğrulayın.

Bu ekran Flutter'ın gerçek widget testinden alınmıştır; kaynak proje çevrimdışı pakette çalışır durumdadır.

Özgün günlük yaşam uygulaması — Etkinlik Başvurusu

Etkinlik Başvurusu

HAFTA 11

Etkinlik Başvurusu

Form, TextFormField, validator ve güvenli veri alma

☰

Deneme verisi

▶ Uygula

Sıfırla

11

Canlı sonuç
Uygulama çalışıyor; farklı girdilerle test edin.

✓

💡

Boş, sınır ve hatalı değerleri de deneyerek sonucu doğrulayın.

Aynı teknik kazanım yeni bir problem bağlamında uygulanır. Öğrenci, örneği kopyalamak yerine veriyi ve ekran geri bildirimini kendi senaryosuna göre değiştirir.

Uygulama zamanı

Kolay	Başlık, renk, simge ve açıklamayı kendi senaryonuza uyarlayın.
Orta	Boş ve geçersiz girdiyi engelleyip Türkçe doğrulama mesajı gösterin.
İleri	Sonuçları bir modele dönüştürüp ListView içinde geçmiş olarak saklayın; silme ekleyin.

Geri bildirimden sonra kontrol: controller kaynakları dispose edildi mi, dar ekranda taşma var mı, boş/hata/yükleniyor durumları gösteriliyor mu ve en az üç test verisi denendi mi?

12. Hafta — Seçim Bileşenleri

Checkbox, Switch, Radio ve DropdownButton

Bu derste konu, önce çalışan bir mobil arayüz üzerinde görülür; sonra widget ağacı küçük parçalara ayrılır. Kaynak kitaptaki kazanım korunur, ancak XML/Activity merkezli sıra yerine Flutter'ın widget, durum ve geri bildirim yaklaşımı kullanılır.

1	Checkbox, Switch, Radio ve DropdownButton
2	Girdi, işlem, çıktı ve durum değişimini ayırt etme
3	Okunabilir Dart kodu ve yeniden kullanılabilir widget geliştirme
4	Boş, sınır ve hatalı verilerle test etme

Çözümlü kitap uyarlaması: Tercih Formu

1. MaterialApp ve Scaffold ile ekran iskeletini kurun. 2. Veriyi TextField ile alın. 3. Değişen değeri State içinde tutun. 4. Olaydan sonra setState ile ekranı yenileyin. 5. Kullanıcıya açık sonuç ve hata mesajı gösterin.

```
FilledButton(onPressed: () => setState(() => value++), child: const Text('Uygula'));
```

Çalışan ekran — Tercih Formu

Tercih Formu

HAFTA 12

Tercih Formu

Checkbox, Switch, Radio ve DropdownButton



Deneme verisi

► Uygula

Sıfırla

12

Canlı sonuç

Uygulama çalışıyor; farklı girdilerle test edin.



Boş, sınır ve hatalı değerleri de deneyerek sonucu doğrulayın.

Bu ekran Flutter'ın gerçek widget testinden alınmıştır; kaynak proje çevrimdışı pakette çalışır durumdadır.

Özgün günlük yaşam uygulaması — Kantin Siparişi

Kantin Siparişi

HAFTA 12

Kantin Siparişi

Checkbox, Switch, Radio ve DropdownButton

☰ Deneme verisi

▶ Uygula

Sıfırla

Canlı sonuç

12 Uygulama çalışıyor; farklı girdilerle test edin.

💡 Boş, sınır ve hatalı değerleri de deneyerek sonucu doğrulayın.

Aynı teknik kazanım yeni bir problem bağlamında uygulanır. Öğrenci, örneği kopyalamak yerine veriyi ve ekran geri bildirimini kendi senaryosuna göre değiştirir.

Uygulama zamanı

Kolay	Başlık, renk, simge ve açıklamayı kendi senaryonuza uyarlayın.
Orta	Boş ve geçersiz girdiyi engelleyip Türkçe doğrulama mesajı gösterin.
İleri	Sonuçları bir modele dönüştürüp ListView içinde geçmiş olarak saklayın; silme ekleyin.

Geri bildirimden sonra kontrol: controller kaynakları dispose edildi mi, dar ekranda taşma var mı, boş/hata/yükleniyor durumları gösteriliyor mu ve en az üç test verisi denendi mi?